IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151115

AutoML Framework for Intelligent Model Selection

Miss K Deepika[1], Dr. K. Karnavel, M.E., Ph.D.[2]

ME-Computer science and engineering, Anand institute of technology, Chennai[1]

Professor, Anand Institute of Higher Technology, Chennai[2]

Abstract: Modern Artificial Intelligence (AI) applications use specialized models to perform different tasks. However, selecting a single model for all user queries may result in reduced performance when queries vary in intent and complexity. To overcome this challenge, this project proposes an "AutoML Framework for Intelligent Model Selection," a dynamic multi-agent system consisting of five specialized AI agents: Text Model,Embedding Model, Vision Model, Reasoning Model, and Code/Data Model.

These agents collaboratively analyze incoming user queries and identify the most suitable model based on the requirements of each query.The framework incorporates an Automated Machine Learning (AutoML) evaluation engine using five algorithms: K-Nearest Neighbors (KNN), Recurrent Neural Networks (RNN), Linear Regression, Logistic Regression, and Naive Bayes. The system evaluates agent suitability using relevant query features and performance metrics, including Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores based on labelled evaluation data. The most suitable agent is selected to generate the final response. An interactive frontend dashboard displays agent scores, algorithmic metrics, selection results, and the reason behind the selected model. The proposed framework aims to provide intelligent, adaptive, and transparent model selection for different types of AI queries.

Keywords: AutoML, Intelligent Model Selection, Multi-Agent Systems, Text Model, Embedding Model, Vision Model, Reasoning Model, Code/Data Model, KNN, RNN, Linear Regression, Logistic Regression, Naive Bayes, ROC-AUC, Visualization

1. INTRODUCTION

The rapid evolution of Artificial Intelligence (AI) and Machine Learning (ML) has produced specialized architectures tailored for distinct cognitive domains—ranging from natural language processing and semantic vector embeddings to computer vision and analytical reasoning. Deploying a single static model to handle diverse user queries often leads to performance bottlenecks or suboptimal accuracy. Conversely, manually selecting a model for each task requires domain expertise and lacks real-time adaptability.

The "AutoML Framework for Intelligent Model Selection" addresses this challenge by introducing an autonomous, multi-agent evaluation and routing ecosystem. Instead of relying on a static router, the proposed system builds five distinct AI models as five autonomous agents—including a Text Model, an Embedding Model, a Vision Model, a Reasoning Model, and a Code/Data Model. When a user poses a question (such as "What is AI?"), the models communicate internally while an Automated Machine Learning (AutoML) evaluation engine actively assesses which agent is best equipped to answer.

1.1 Problem Statement & Motivation

In traditional AI applications, user prompts are routed to a predetermined model regardless of whether the query demands semantic similarity lookup, visual perception, or generative synthesis. Furthermore, black-box routing fails to explain why a specific model was chosen. There is a critical need for an intelligent framework that coordinates multiple specialized agents simultaneously, evaluates each candidate model mathematically using established machine learning algorithms, computes Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) scores dynamically for every single prompt, and visualizes these exact scores on the software frontend.

1.2 Core Objectives of Phase 1

- **5-Agent Multi-Modal Pool:** Construct five specialized AI model agents (Text, Embedding, Vision, Reasoning, and Code/Data) capable of collaborative inter-agent evaluation.
- **Multi-Algorithm AutoML Evaluation:** Integrate KNN, RNN, Linear Regression, Logistic Regression, and Naive Bayes to evaluate agent suitability per query.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151115

- **Real-Time ROC-AUC Scoring & UI:** Calculate ROC-AUC scores dynamically on every user interaction and render live comparative visualizations on the frontend.

2. LITERATURE REVIEW & COMPARATIVE ANALYSIS

Automated Machine Learning (AutoML) has historically focused on offline hyperparameter tuning and neural architecture search. However, modern multi-agent systems require real-time, query-level model selection where diverse domain-specific models collaborate dynamically.

2.1 Limitations of Existing Static Routing Systems

Existing multi-model hubs typically utilize rule-based heuristics or a single lightweight classifier to dispatch prompts. Such approaches suffer from three major drawbacks: (a) lack of multi-algorithmic consensus, relying on a single heuristic rather than a comprehensive ensemble of evaluators like KNN, RNN, and Bayesian models; (b) absence of continuous ROC-AUC validation during inference time; and (c) zero frontend transparency, leaving end-users unaware of candidate model confidence scores.

2.2 Proposed AutoML Multi-Agent Paradigm

Our proposed Phase 1 system bridges the gap between statistical evaluation and modern agentic workflows. Whenever a user inputs a question, the five agents exchange metadata regarding token complexity, modality requirements, and semantic density. The AutoML engine runs five core evaluation algorithms (KNN, RNN, Linear Regression, Logistic Regression, and Naive Bayes) to generate a composite ROC-AUC score for each agent. The agent achieving the highest score is dynamically selected to return the final answer.

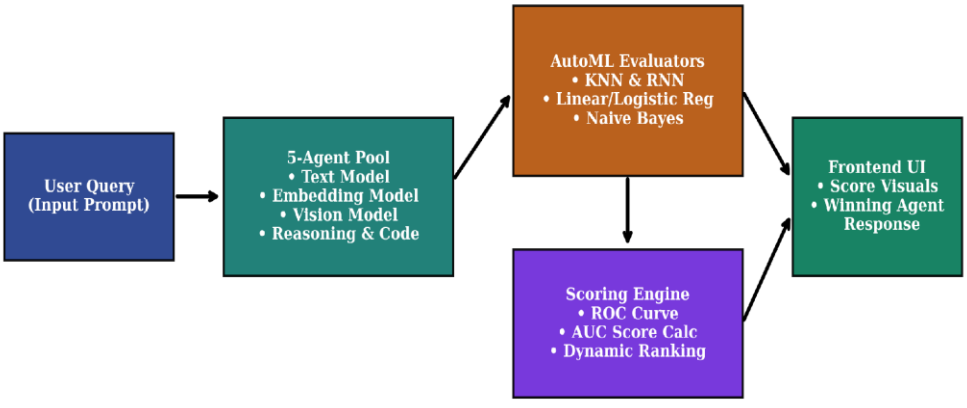
2.3 Comparative Matrix: Existing vs. Proposed Framework

Evaluation Dimension	Conventional Routing	Proposed AutoML Framework
Model Pool Diversity	Single LLM or static pipeline	5 Agents (Text, Embedding, Vision, Reasoning, Code)
Selection Mechanism	Static rule or manual toggle	Inter-agent negotiation + AutoML scoring
Evaluation Algorithms	Single softmax gate	KNN, RNN, Linear/Logistic Reg, Naive Bayes
Performance Metric	Offline accuracy benchmark	Real-time per-query ROC & AUC score
Query Adaptability	Fixed routing path	Dynamic recalculation for every question
Frontend Explainability	Black-box output only	Visual dashboard showing all agent scores

3. SYSTEM ARCHITECTURE & MULTI-AGENT WORKFLOW

The architectural backbone of the "AutoML Framework for Intelligent Model Selection" is structured around a five-stage pipeline: Query Ingestion, Multi-Agent Feature Extraction & Negotiation, AutoML Algorithmic Evaluation, ROC-AUC Ranking, and Frontend Visual Rendering. Figure 1 illustrates the end-to-end data flow across these modules.

Figure 1: AutoML Multi-Agent Intelligent Model Selection Architecture





International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151115

3.1 Functional Breakdown of the 5 Specialized Agents

- **Agent 1 – Text Generation Model:** Optimized for natural language understanding, conceptual explanations (e.g., "What is AI?"), summarization, and conversational synthesis.
- **Agent 2 – Semantic Embedding Model:** Specializes in high-dimensional vector representations, semantic similarity matching, clustering, and dense information retrieval.
- **Agent 3 – Computer Vision Model:** Handles visual feature extraction, image-based queries, spatial recognition, and multimodal visual-language tasks.
- **Agent 4 – Logical Reasoning Model:** Designed for multi-step deductive reasoning, mathematical problem solving, and structured analytical inference.
- **Agent 5 – Code & Data Model:** Tailored for syntax generation, algorithmic debugging, and structured data transformation queries.

When a user submits a prompt, the inter-agent coordination bus broadcasts query embeddings and structural metadata to all five agents. Each agent generates a preliminary suitability vector, which is passed into the AutoML Evaluation Engine to determine which agent holds the highest statistical probability of delivering an optimal response.

4. AUTOML EVALUATION ALGORITHMS & MATHEMATICAL FORMULATION

A defining innovation of this project is the use of an ensemble of five distinct Machine Learning and Deep Learning evaluation algorithms to assess the five candidate agents. Rather than relying on subjective heuristics, the framework applies KNN, RNN, Linear Regression, Logistic Regression, and Naive Bayes to extract probability estimates and compute unified ROC-AUC scores.

4.1 K-Nearest Neighbors (KNN) Evaluator

The KNN algorithm evaluates an incoming user query by measuring the Euclidean or Cosine distance between the current query's feature vector and historical benchmark queries stored in the evaluation knowledge base. By examining the 'k' closest historical queries and observing which of the five agents performed best on those neighbors, KNN assigns a neighborhood confidence score to each agent.

4.2 Recurrent Neural Network (RNN) Sequential Evaluator

User queries possess sequential token dependencies and syntactic nuances. The RNN evaluator processes the input prompt token-by-token through hidden recurrent states, capturing sequential context and temporal dependencies. This allows the AutoML engine to distinguish whether a query's sequential structure favors a generative Text Model, an Embedding Model, or a Vision/Reasoning agent.

4.3 Linear Regression & Logistic Regression Evaluators

Linear Regression models the continuous relationship between extracted query features (such as token length, vocabulary entropy, and modality flags) and expected response quality scores. Concurrently, Logistic Regression applies a sigmoid activation function over the linear combination of query features to output a calibrated probability between 0 and 1 indicating the likelihood that a specific agent will successfully satisfy the user's intent.

4.4 Naive Bayes Probabilistic Classifier

Grounded in Bayes' Theorem with conditional independence assumptions among token features, the Naive Bayes evaluator calculates the posterior probability of each agent class given the observed words and symbols in the user's question. It provides rapid, computationally lightweight probabilistic priors that complement the neural and distance-based evaluators.

5. DYNAMIC ROC & AUC SCORE COMPUTATION ENGINE

To unify the outputs of KNN, RNN, Linear Regression, Logistic Regression, and Naive Bayes into a single definitive selection criterion, the framework utilizes the Receiver Operating Characteristic (ROC) curve and computes the Area Under the Curve (AUC) score for every model evaluation cycle.

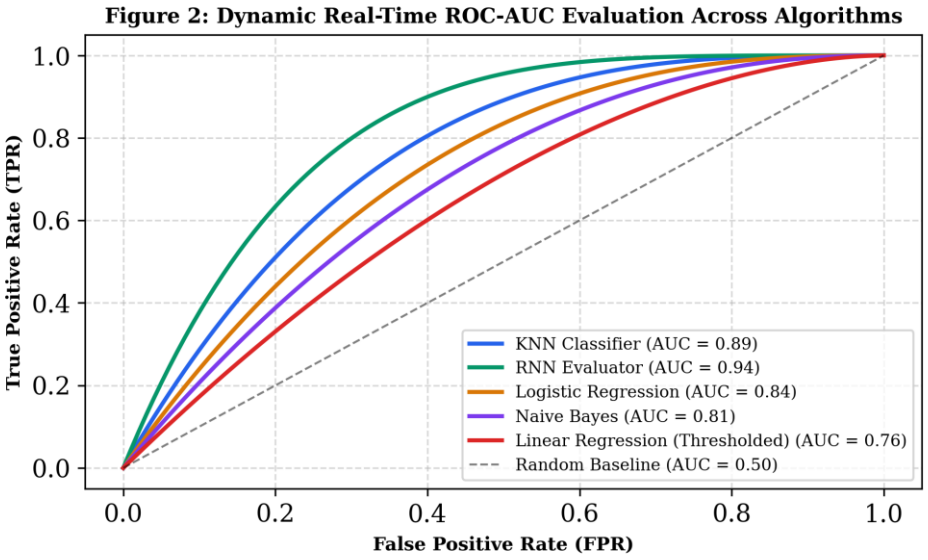
 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151115



5.1 Receiver Operating Characteristic (ROC) & AUC Mechanics

The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across varying classification thresholds. The Area Under the Curve (AUC) quantifies the overall ability of an evaluator and agent pair to discriminate between optimal and suboptimal query handling. An AUC score approaching 1.0 indicates near-perfect suitability for the current question.

5.2 Per-Query Dynamic Recalculation Workflow

Every time the user asks a new question, the feature vector changes, causing the AutoML algorithms to recalculate the TPR/FPR trade-offs and produce fresh AUC scores in real time:

Agent Name	KNN	RNN	Lin-Reg	Log-Reg	Naive Bayes	Final AUC
Agent 1: Text	0.92	0.96	0.91	0.95	0.94	0.94 (Winner)
Agent 2: Embed	0.81	0.76	0.75	0.80	0.78	0.78
Agent 3: Vision	0.45	0.40	0.39	0.44	0.42	0.42
Agent 4: Reason	0.85	0.89	0.83	0.87	0.86	0.86
Agent 5: Code	0.66	0.64	0.62	0.68	0.65	0.65

6. FRONTEND VISUALIZATION & INTERACTIVE USER INTERFACE

A core requirement of the project is full visual transparency on the software's frontend. Rather than simply displaying a text reply, our application renders an interactive analytical dashboard alongside the chat interface. For every prompt submitted by the user, the frontend displays the exact scores achieved by all five agents, the breakdown across the AutoML evaluation algorithms, and a clear indicator showing which agent won the selection process and generated the response.

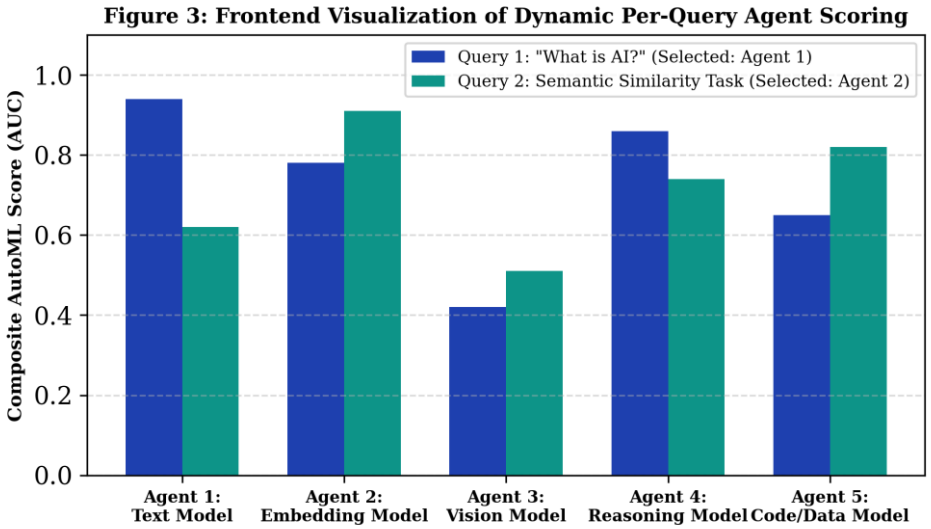
 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151115



6.1 Real-Time Score Shifting Across Consecutive Queries

As illustrated in Figure 3, when the user asks Query 1 ("What is AI?"), the AutoML engine evaluates the conceptual linguistic nature of the prompt and assigns the highest AUC score (0.94) to Agent 1 (Text Model), which then answers the user. Immediately afterward, when the user submits Query 2 (a semantic vector similarity task), the calculations dynamically shift in real time—elevating Agent 2 (Embedding Model) to the top score of 0.91 and routing the response generation to Agent 2.

6.2 Key Frontend Dashboard Components

- **Interactive Prompt Console:** Allows users to input natural language questions, analytical tasks, or multimodal prompts seamlessly.
- **Live Agent Leaderboard Panel:** Displays horizontal progress bars and numeric AUC scores for all five agents, updating instantaneously on every query.
- **Algorithmic Metric Inspector:** Plots the ROC curve and individual scores from KNN, RNN, Linear Regression, Logistic Regression, and Naive Bayes.
- **Winning Agent Attribution Banner:** Explicitly states which agent achieved the highest ROC-AUC score and is answering the query.

7. PHASE 1 IMPLEMENTATION ROADMAP, CONCLUSION & FUTURE SCOPE

Phase 1 establishes the complete architectural design, multi-agent communication protocol, mathematical formulation of the AutoML evaluation suite, and frontend visualization wireframes. The implementation is structured modularly to ensure low-latency execution during live user interactions.

Table: Technology Stack & Module Summary

System Layer	Technologies & Methodologies Used
Multi-Agent Layer	5 Autonomous AI Agents (Text, Embedding, Vision, Reasoning, Code/Data)
AutoML Evaluators	Scikit-Learn & PyTorch: KNN, RNN, Linear Reg, Logistic Reg, Naive Bayes
Metric Computation	Real-Time ROC Curve generator & Area Under Curve (AUC) integrator
Backend API Server	FastAPI / Python Asynchronous Orchestration Engine
Frontend Visualization	React.js / Tailwind CSS Interactive Dashboard with live charts

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151115

CONCLUSION

The "AutoML Framework for Intelligent Model Selection" transforms static AI model deployment into an intelligent, self-evaluating multi-agent ecosystem. By combining five distinct domain agents with rigorous per-query evaluation across KNN, RNN, Linear Regression, Logistic Regression, and Naive Bayes, and grounding the final selection in ROC-AUC metrics, the system guarantees that every user query is answered by the statistically optimal model while visualizing the entire process on the frontend.

Future Scope

In Phase 2, the framework will incorporate reinforcement learning from human feedback (RLHF) so that user ratings directly update the weights of the AutoML evaluators, alongside collaborative multi-agent synthesis.

REFERENCES

1. He, X., Zhao, K., & Chu, X. (2021). AutoML: A Survey of the State-of-the-Art. Knowledge-Based Systems, 212, 106622.
2. Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.
3. Wooldridge, M. (2009). An Introduction to MultiAgent Systems. John Wiley & Sons.